网络爬虫1

1 网络爬虫

定义：

网络蜘蛛，网络机器人，抓取网络数据的程序

总结：

用Python程序去模仿人去访问网站，模仿的越逼真越好

目的：

通过有效的大量的数据分析

2 企业获取数据的方式

1.公司自有

2.第三方数据平台购买

3 Python爬虫优势

python：

请求模块，解析模块丰富成熟

php：

对多线程，异步支持不够好

JAVA：

代码量大，代码笨重

C/C++：

效率高，但是代码成型慢

4 爬虫分类

4.1 通用网络爬虫（搜索引擎引用，需要遵守rotbots协议）

http://www.qq.com/rotbots.txt

1.搜索引擎如何获取一个新网站的URL

网站主动向搜索引擎提供（百度站长平台）

2.和DNS服务商（万网），快速收录新网站

4.2 聚焦网络爬虫

自己写的爬虫程序：面向主题的爬虫、面向需求的爬虫

5 爬取数据步骤

1.确定需要爬去的URL地址

2.通过HTTP/HTTPS协议来获取相应的HTML页面

3.提取HTML页面里有用的数据

①所需数据，保存

②页面中其他的URL，继续执行第2步

6 Chrome浏览器插件

6.1 插件安装步骤

1.设置---更多工具---扩展程序

2.点开开发者模式

3.把插件拖拽到浏览器界面

6.2插件介绍

1.Proxy SwitchOmega：代理切换插件

2.XPath：网页数据解析插件

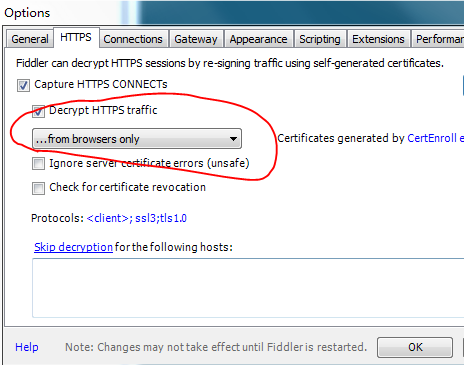
3.JSON View：查看json格式的数据（好看）

7 Filldler抓包工具

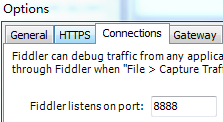
7.1 抓包设置

1.设置Filldler抓包工具

Tools---Options---Actions



Actions---Trust Root certificate

插件设置端口和Fiddler listens on port一致

2.设置浏览器代理

8 Anaconda和Spyder

1.anaconda：开源的Python发行版本

2.Spyder：继承的开发环境

Spyder常用快捷键

1.注释 ctrl+L

2.运行程序 F5

9 WEB

9.1 HTTP和HTTPS

1.HTTP:80

2.HTTPS:443 HTTP的升级版

9.2 GET和POST

1.GET:查询参数会在URL上显示出来

2.POST：查询参数和提交数据在form表单里，不会再URL地址栏

9.3 URL

http://item.jd.com :80/ 26606127795.html #detail

协议 域名/IP地址 端口号 资源路径 锚点(类似于商品详情)

9.4 User-Agent

记录用户的浏览器，操作系统等，为了让用户获取更好的HTML页面效果

10 爬虫请求模块

urllib.request

10.1 版本

Python2：urllib和urllib2

Python3：把两者合并，urllib.request

10.2 常用方法

urllib.request.urlopen(“url”)

作用：

向网站发起请求并获取响应

#URLopen()得到的响应对象response是bytes类型

#response.read().decode(“utf-8”)转换为str



urllib.request.Request(url,headers={})

作用：

重构User-Agent，爬虫与反爬虫斗争第一步

使用步骤：

1.构建请求对象request：Request()

2.获取响应对象response：urlopen(request)

3.利用响应对象response.read().decode(‘utf-8’)



10.3 请求对象request的方法

|  |  |
| --- | --- |
| 方法 | 作用 |
| request.add\_header() | 添加或者修改headers(User-Agent) |
| get\_header(‘User-agent’) | 获取已有的HTTP报头的值 |



10.4 响应对象response方法

|  |  |
| --- | --- |
| read() | 读取服务器响应内容 |
| getcode() | 返回HTTP的响应码  200成功 4xx服务器页面出错  5xx服务器出错 |
| info() | 返回服务器报头信息 |

urllib.parse

|  |  |
| --- | --- |
| quote(中文) | 编码 |
| unquote(编码之后的字符串) | 解码 |
| urlencode(字典) | 字典编码 |



2.urlencode(字典)

url：wd=’美女’

d = {

“wd”:”美女”

}

d = urllib.parse.urlencode(d)



练习：

百度贴吧抓取

要求：

1、输入贴吧名称

2、输入抓取的起始页和终止页码

3、把每一页的内容保存到本地：第一页.html 第二页.html

步骤：

1、找URL规律（拼接URL）

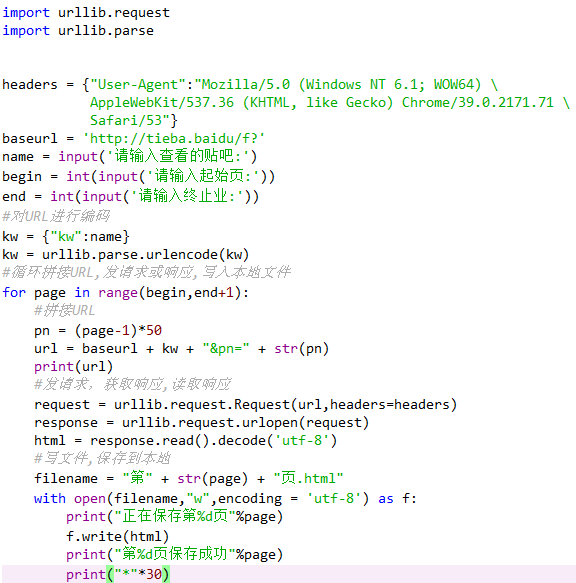
第一页:http://www.tieba.baidu.com/f?kw=达内&pn=0

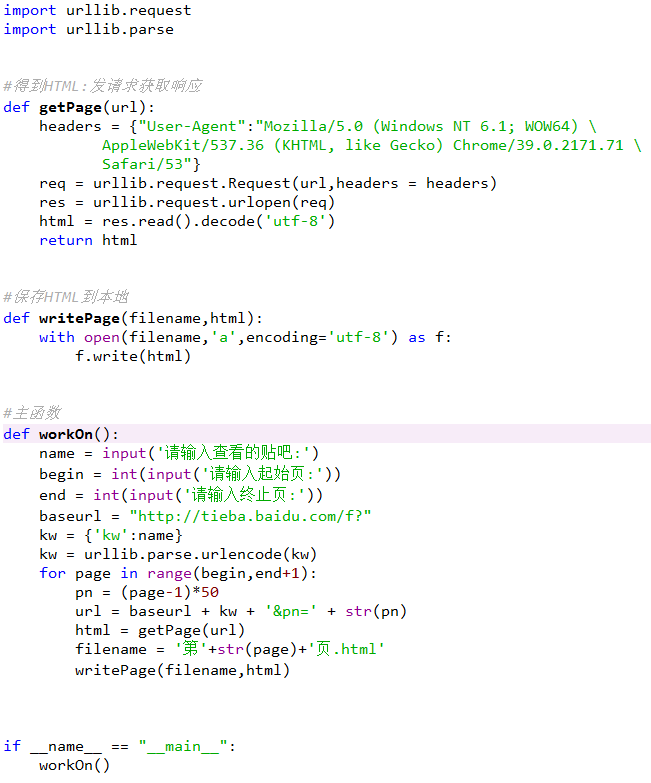
第二页:http://www.tieba.baidu.com/f?kw=达内&pn=50

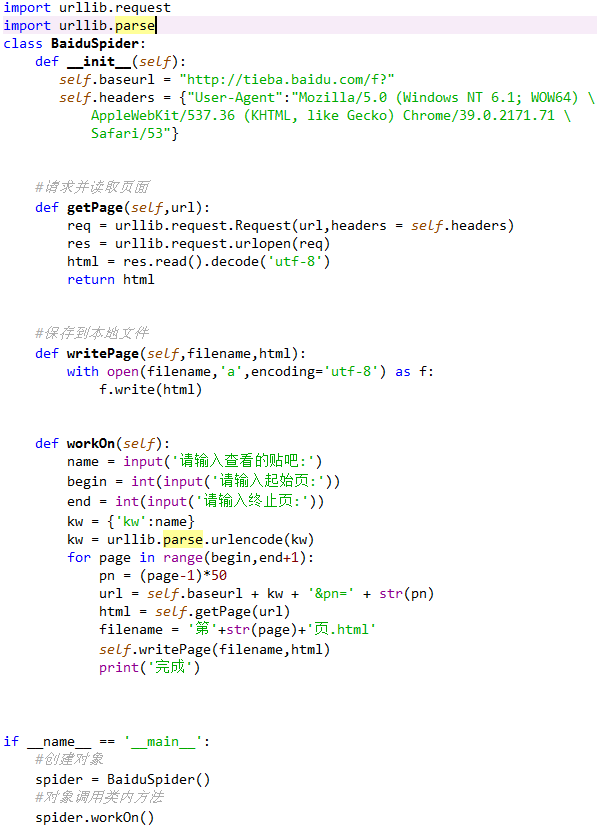
第三页:http://www.tieba.baidu.com/f?kw=达内&pn=100

2、获取响应内容

3、保存到本地\数据库







作业：

1.爬取猫眼电影top100榜单

①程序运行，直接爬取第一页

②提示：是否继续爬取(y/n)，y爬取第二页以此类推，n显示爬取结束

③把每一页的内容保存到本地，第一页.html

步骤：

1.URL

2.获取响应内容

3.写入到本地

网络爬虫2

1 解析

1.1 数据分类

结构化数据

特点

有固定的格式：HTML、XML、JSON等

非结构化数据

示例：

图片、音频、视频这类数据一般存储为二进制

with open(‘gril.jpg’,’wb’)as f:

f.write(html)

2 正则表达式re

2.1 使用流程

1.创建使用对象：p = re.compile(‘\d’)

2.对字符串匹配：result = p.match(“123ABC”)

3.获取匹配结果：print(result.group())

2.2 常用方法

1.match(s)：只匹配字符串开头，返回对象

2.search(s)：从开始到结束去匹配，匹配到之间返回对象

3.group()：从match和search返回的对象中取值

4.findall(s)：全部匹配，返回一个列表

2.3 表达式

. 匹配任意字符（不能匹配\n）

[…]包含中括号中的内容 A[BCD]E结果ABE，ACE，ADE

\d数字

\w 字母、数字、下划线

\s空白字符

\S非空白字符

\*前一个字符出现多次或零次

？前一个字符出现零次或者一次

+前一个字符出现一次或者多次

{m}前一个字符出现m次

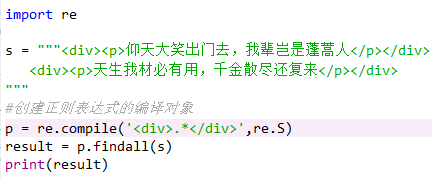
贪婪匹配：

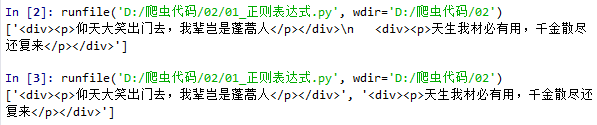
在整个表达式匹配成功的前提下，尽可能多的去匹配\*

非贪婪匹配：

在整个表达式匹配成功的前提下，尽可能少的去匹配\*

贪婪模式，非贪婪加一个?号





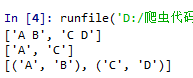
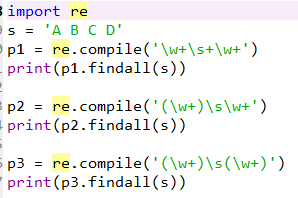
示例：

“””<div><p>仰天大笑出门去，我辈岂是蓬蒿人</p></div>

<div><p>天生我材必有用，千金散尽还复来</p></div>

“””

2.4 findall()的分组



匹配后只显示括号里的

练习：

<div class=’animal’>

<p class=’name’>

<a title=’Tiger’>

</a>

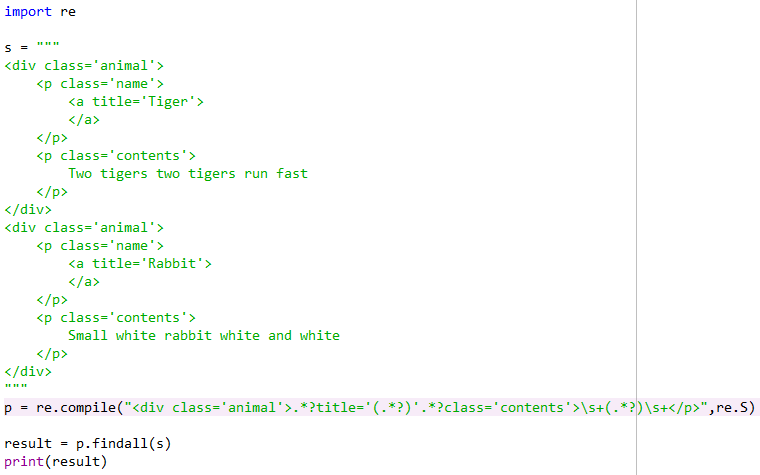
</p>

<p class=’contents’>

Two tigers two tigers run fast

</p>

</div>







2.5 抓取猫眼电影TOP100

1.猫眼电影 - top100榜单

2.目标：电影名、主演、上映时间

3.步骤

①找URL

第一页：<http://maoyan.com/board/4?offset=0>

第二页：<http://maoyan.com/board/4?offset=10>

②用正则匹配相应内容

<div class="movie-item-info">.\*?title="(.\*?)".\*?class="star">(.\*?)</p>.\*?class="releasertime">(.\*?)</p>.\*?</div>

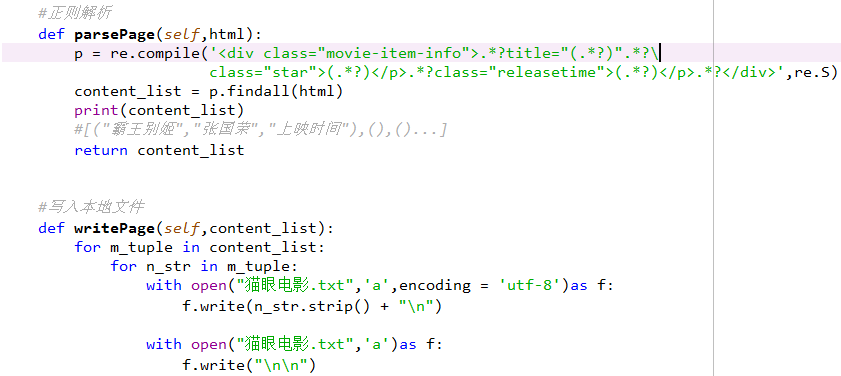
③写代码

1、获取每页HTML源码

2、正则处理HTML源码

3.写入本地文件





3 Fiddler抓包工具

3.1 抓包设置

设置Fiddler抓包工具

HTTPS：Tools--->HTTPS--->…from browsers only--->Actions证书信任

connections：设置端口号8888

设置浏览器代理

Proxy SwitchOmega-选项---新建情景模式---HTTP 127.0.0.1 8888

3.2 常用菜单

Inspectors：查看抓到的数据包的详细内容

常用选项：

Headers：客户端发送到服务器的请求头header，包含了web客户端信息，cookie，传输状态

WebForm：显示请求POST的数据

Raw：将整个请求显示为纯文本

4 请求方式及其案例

GET（查询参数都在URL地址中显示）

POST

特点：查询参数都在Form表单里保存

使用：urllib.request.urlopen(url,data=formdata,headers=headers)

data：表单数据必须为bytes数据类型提交

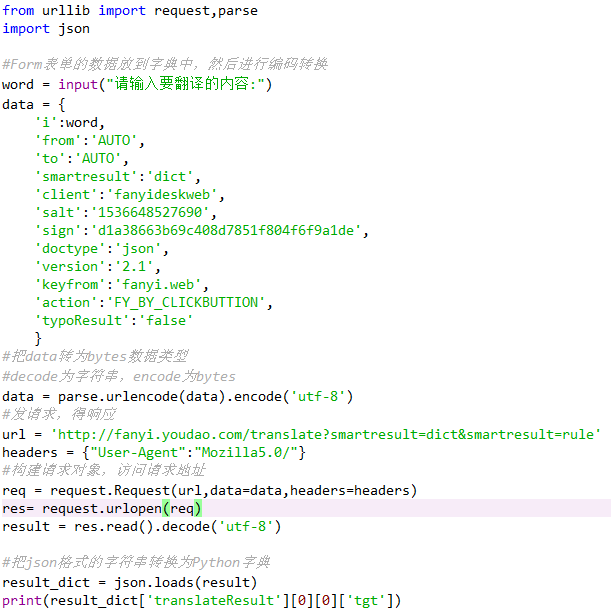
4.1 案例：有道翻译

1.利用Fiddler抓包工具抓取Form表单数据

2.对POST数据进行处理，转换成bytes数据类型

3.发请求得响应

decode为字符串，encode为bytes



4.2 Cookie模拟登陆

1.cookie和session

cookie：通过在客户端记录的信息确定用户身份

session：通过在服务器端记录的信息确定用户身份

案例：使用cookie登陆人人网

5 json模块

json.loads(“json格式的字符串”)

作用：

把json格式的字符串--->Python字典

5 requests模块

5.1 安装模块

pip3 install requests

1.安装(Conda prompt终端)

(base) > conda install requests

或者

(base)>pip install requests

5.2 使用方法

get()

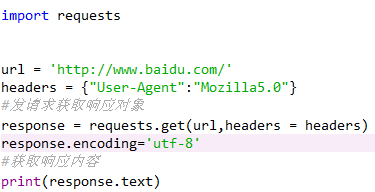
向网站发起请求并获取响应对象

用法：

get(url,headers=headers)

response属性

|  |  |
| --- | --- |
| response.text | 获取响应内容（字符串）返回类型ISO-8859-1  手动指定字符编码：response.encoding=’utf-8’ |
| response.content | 获取响应内容（bytes数据类型，字节流）  爬取图片、音频、视频等非结构化数据 |
| response.status\_code | 返回服务器的响应码 |

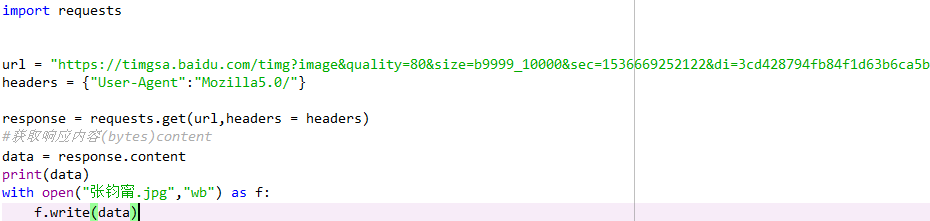


**5.2.1使用response.content爬取图片**

爬取百度美女图片

步骤：

1.找到url



作业：

爬取内涵段子

网址:http://www.neihan8.com

段子下的内涵段子

# -\*- coding: utf-8 -\*-

"""

Created on Tue Sep 11 11:18:07 2018

@author: Administrator

"""

import re

import urllib.request

import urllib.parse

class CatSpider:

def \_\_init\_\_(self):

self.headers = {"User-Agent":"Mozilla/5.0 (Windows NT 6.1; WOW64) \

AppleWebKit/537.36 (KHTML, like Gecko) Chrome/39.0.2171.71 \

Safari/53"}

self.baseurl = 'http://maoyan.com/board/4?offset='

self.page = 1

self.offset = 0

#获取HTML源码

def getPage(self,url):

request = urllib.request.Request(url,headers = self.headers)

response = urllib.request.urlopen(request)

html = response.read().decode('utf-8')

return html

#正则解析

def parsePage(self,html):

p = re.compile('<div class="movie-item-info">.\*?title="(.\*?)".\*?\

class="star">(.\*?)</p>.\*?class="releasetime">(.\*?)</p>.\*?</div>',re.S)

content\_list = p.findall(html)

print(content\_list)

#[("霸王别姬","张国荣","上映时间"),(),()...]

return content\_list

#写入本地文件

def writePage(self,content\_list):

for m\_tuple in content\_list:

for n\_str in m\_tuple:

with open("猫眼电影.txt",'a',encoding = 'utf-8')as f:

f.write(n\_str.strip() + "\n")

with open("猫眼电影.txt",'a')as f:

f.write("\n\n")

#主函数

def workOn(self):

while True:

url = self.baseurl + str(self.offset)

print("正在爬取第%d页" %self.page)

html = self.getPage(url)

content\_list = self.parsePage(html)

self.writePage(content\_list)

print('第%d页爬取成功' %self.page)

choice = input("是否继续爬取下一页(y/n)")

if choice.strip().lower() == 'y':

self.offset += 10

self.page += 1

else:

print("程序结束")

break

if \_\_name\_\_ == "\_\_main\_\_":

spider = CatSpider()

spider.workOn()

5.3 get查询参数

get()查询参数params（字典格式）

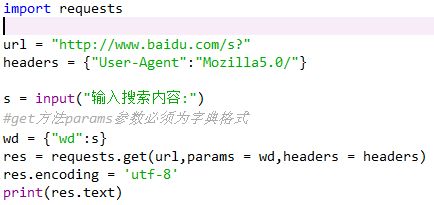
1.没有查询参数

res = requests.get(url,headers=headers)

2.有查询参数

params={‘wd’:’达内’}

res = requests.get(url,params=params,headers=headers)



post()

参数名：data

data = {}#data参数为字典，不用转为bytes数据类型

见示例：



6 代理proxies

6.1 爬虫与反爬虫的第二步

获取代理ip的网站：

西刺代理

快代理

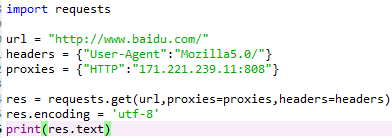
全网代理

1.普通代理：

proxies = {“协议”:”ip地址:端口号”}

proxies = {“HTTP”:” 171.221.239.11:808”}

proxies = random.choice(proxy\_list)



2.私密代理

proxies={“协议”:”http://用户名:密码@ip地址：端口”}

6.2 爬取链家地产二手房信息

目标：爬取小区名称和总价

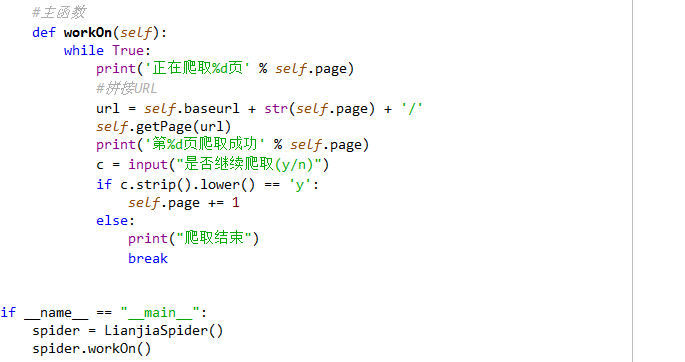
步骤：

1.找URL

<https://xa.lianjia.com/ershoufang/pg1>

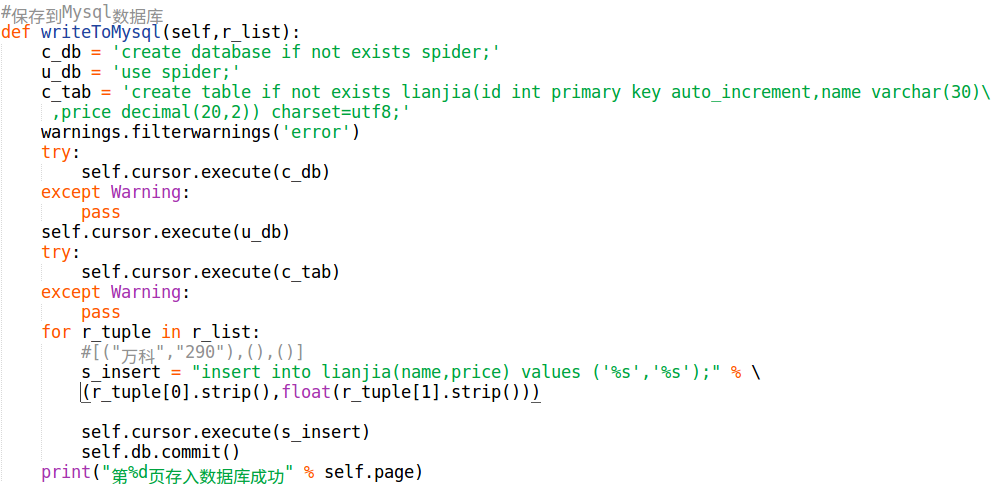
2.正则匹配

3.写入本地文件

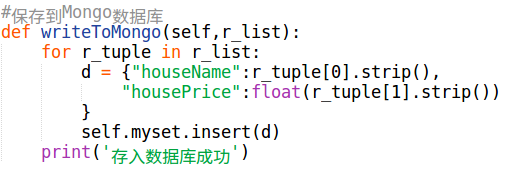
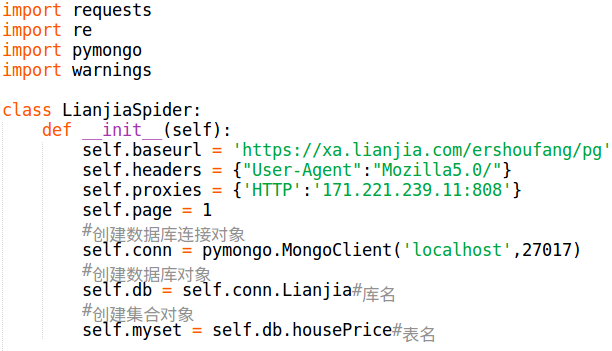


================================存入数据库mysql=====================





============================存入mongo数据库==========================



6.3 Web客户端验证：auth

1.author= (“用户名”,”密码”)

2.爬取<http://code.tarena.com.cn>

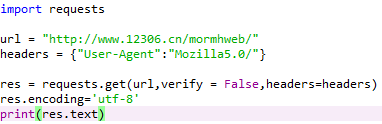
正则：(’<a href=”\w+/”>(.\*?)</a>’,re.S)



6.4 SSL证书认证：verify

1.verify=True：默认，做SSL证书认证

2.verify=False：忽略证书认证



7 Handler处理器（urllib.request）

自定义的urlopen()方法，urlopen方法是一个特殊的opener

7.1 常用方法

①build\_opener(Handler处理器对象)

②opener.open(url) 相当于urlopen()

7.2 Handler使用流程

1.创建相关的Handler处理器对象

http\_handler = urllib.request.HttpHandler({“”：“”})

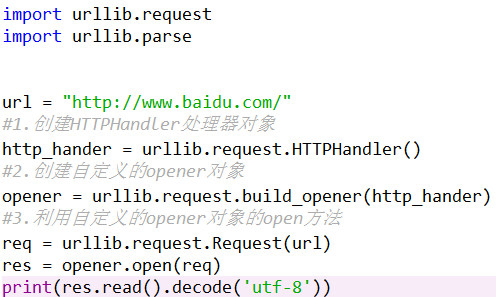
2.创建自定义的opener对象

opener = urllib.request.build\_opener(http\_handler)

3.利用opener对象的open方法发请求

7.3 Handler处理器的分类

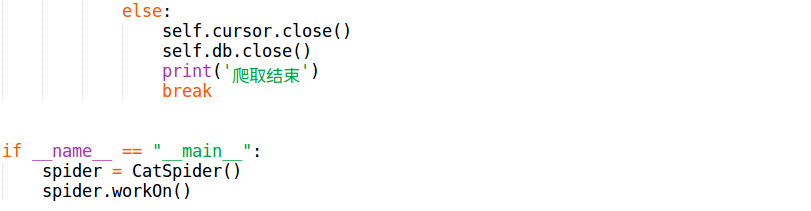
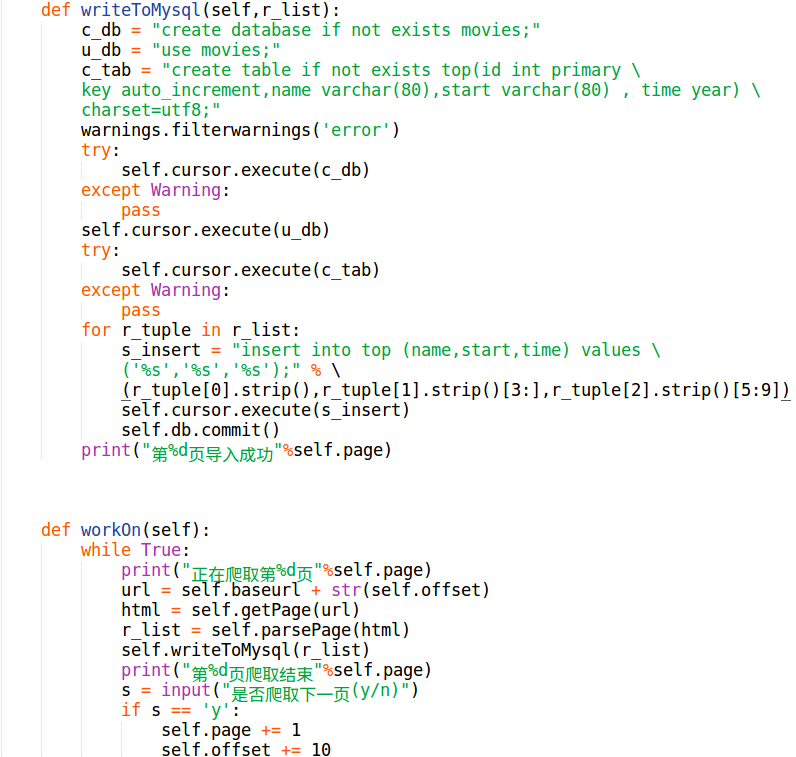
HTTPHandler()



ProxyHandler(代理IP):普通代理

ProxyBasicAuthHandler(密码管理器对象):私密代理





网络爬虫4

1 CSV模块使用流程

1.1 csv模块使用流程

1.Python语句打开CSV文件

with open (“测试.csv”,”a”) as f :

pass

2.初始化写入对象

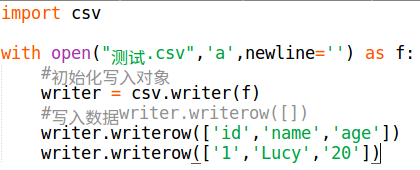
writer方法：

write = CSV.write(f)

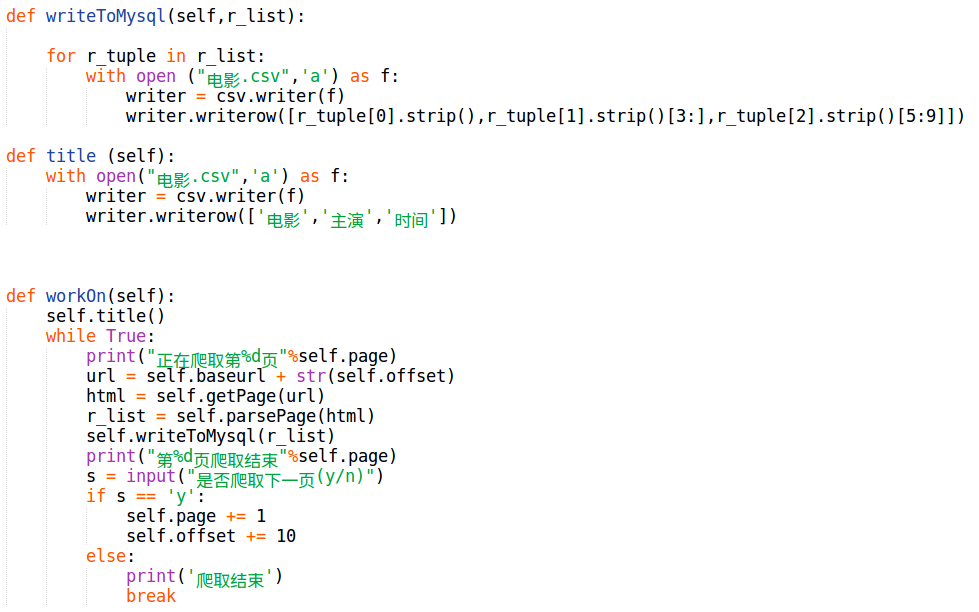
3.写入数据

writerow()方法：

write.writerow([])



===========================猫眼电影=============================



2 Xpath工具

ctrl+shift+x

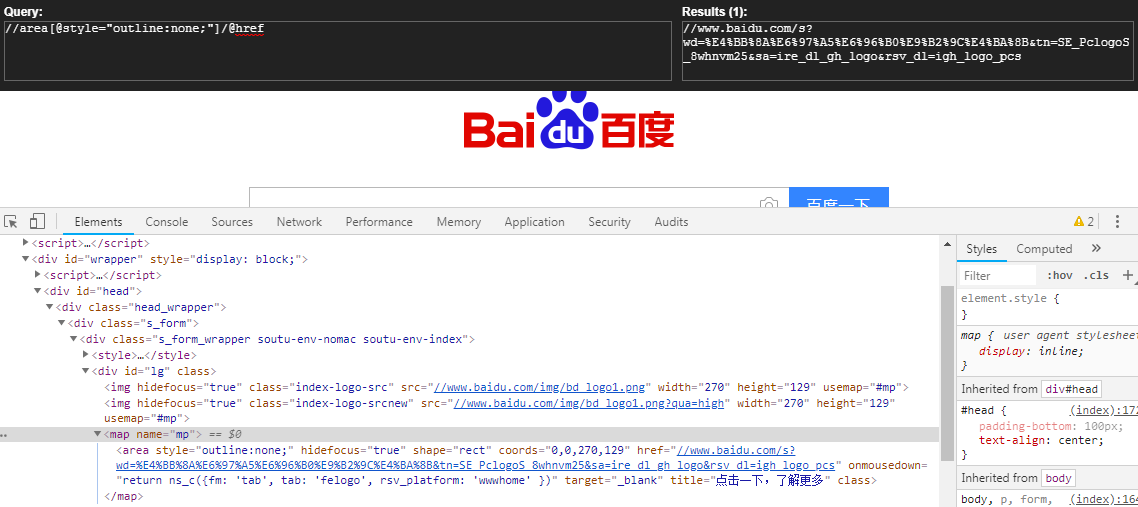
在XML文档中查找信息的语言，同样适用于HTML文档的间检索

2.1 Xpath辅助工具

Chrome插件，XPath Helper

打开/关闭：ctrl+shift+大写X

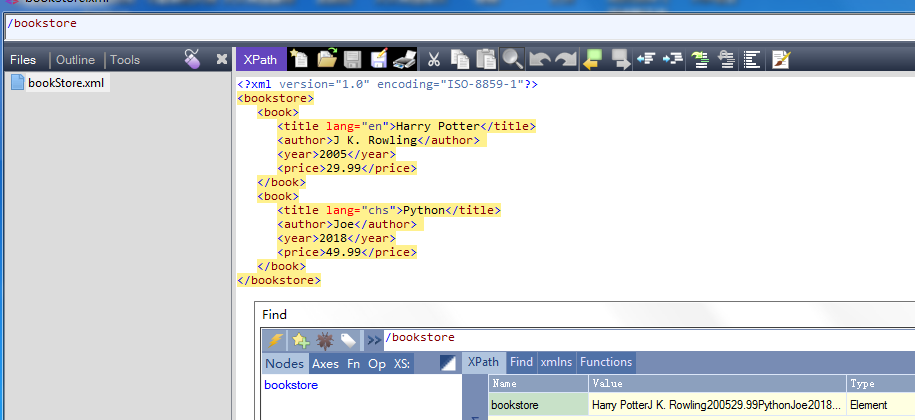
FireFox插件，XPath checker



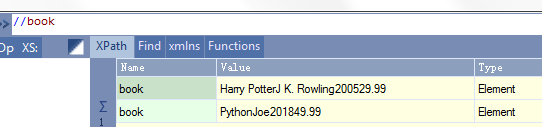
3 Xpath工具

匹配演示

查找bookstore下面所有节点：/bookstore

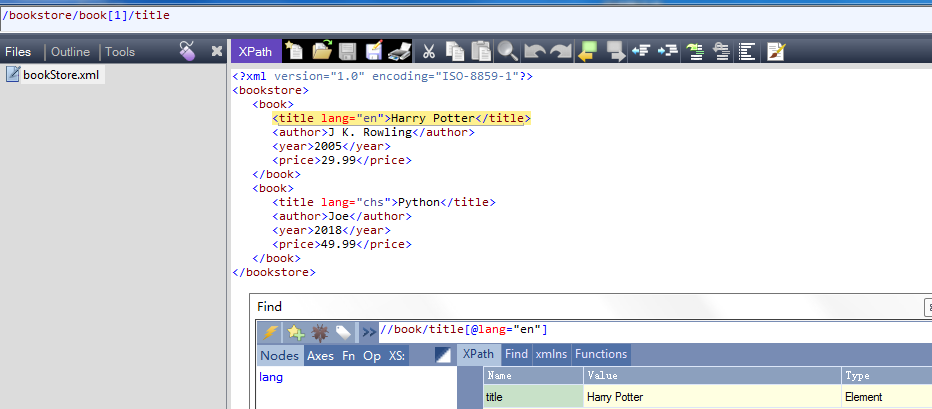


查找所有book节点：//book



查找所有book节点下的title节点中，lang属性为”en”的节点：

//book/title[@lang=”en”]



3.1 选取节点

|  |  |
| --- | --- |
| / | 从根节点开始选取/bookstore(当前节点下) |
| // | 从整个文档中查找某个节点//price（忽略层级关系） |
| @ | 选取某个节点的属性//title[@lang=”en”] |

@使用：

1.选取1个节点：//title[@lang=”en”]

2.选取n个节点：//title[@lang]

3.选取某个属性的值：//title/@lang

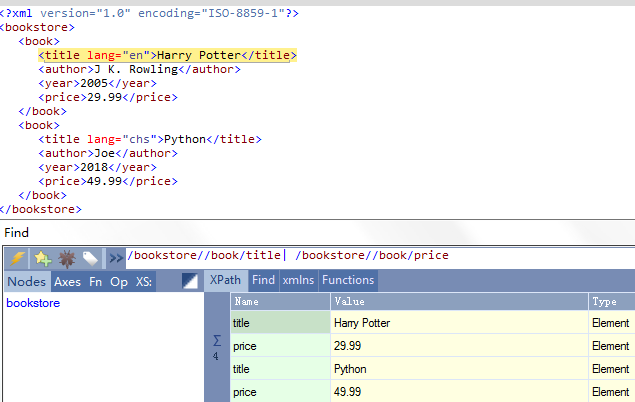
3.2 匹配多路径

符号：

或 |

示例：

获取所有book节点下的title节点和price节点



3.3 函数

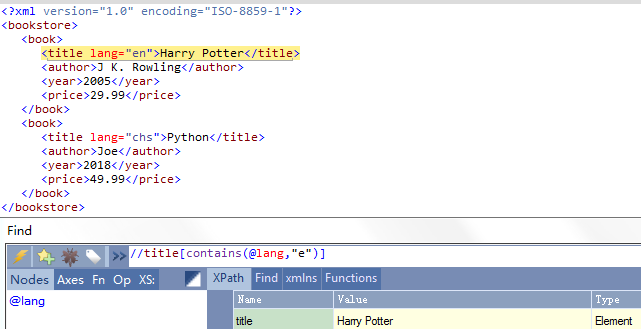
contains()：匹配1个属性值中包含某些字符串的节点

lang=”en”

lang=”chs”

lang=”english”

在所有属性中查找属性值中包含”e”的



4 解析HTML源码

4.1 lxml：HTML/XML解析库

1.安装

conda install lxml

pip install lxml

2.使用流程

1.利用lxml库的etree模块构建解析对象

2.解析对象调用xpath工具定位节点

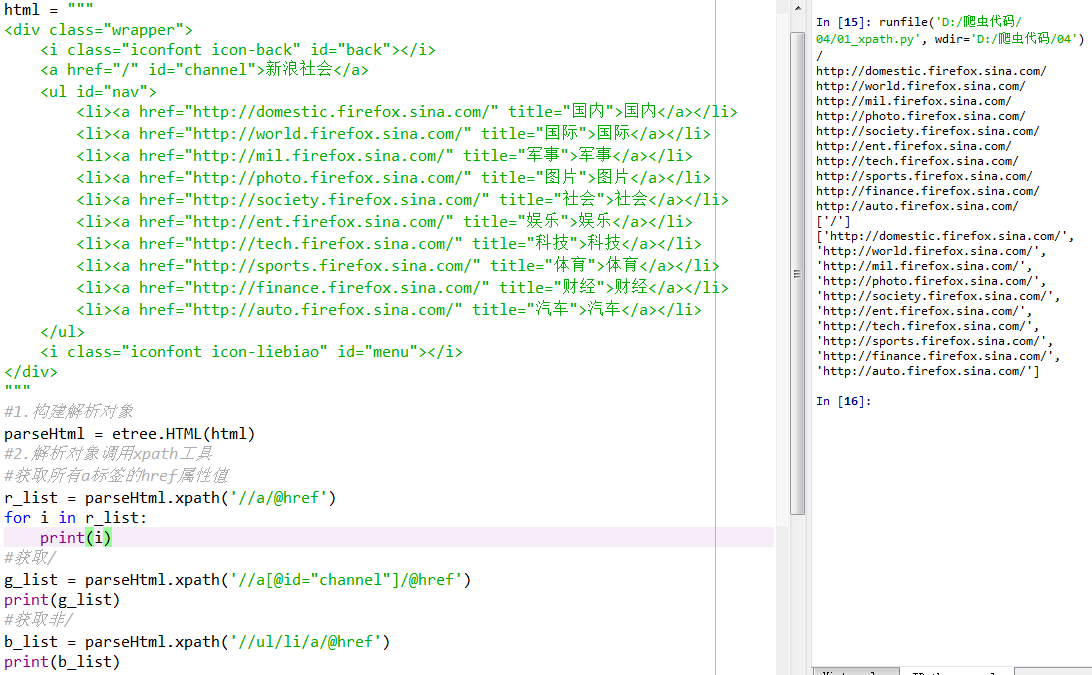
3.使用：

导入模块：from lxml import etree

创建解析对象：parseHtml = rtree.HTML（html）

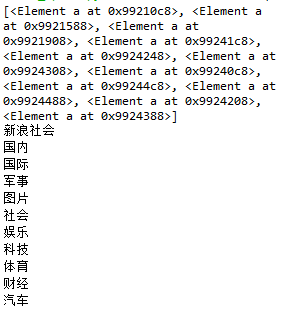
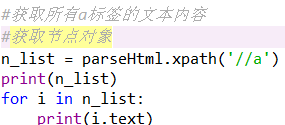
调用xpath进行解析：r\_list = parseHtml.xpath(‘//title[@lang=”en”]’)

#只要调用就一定是列表



==========================获取元素中的内容==========================

对象名.text



4.2 案例

抓取百度贴吧帖子里的图片

1.目标：抓贴吧中帖子图片

2.思路

①获取到贴吧主页的url找到url规律

②获取某个贴吧每个帖子的url

③对每个帖子发送请求，获取帖子里所有图片的url

④对图片url发请求，以wb的方式写入本地文件

3.步骤

<https://tieba.baidu.com/f?kw=%E7%BE%8E%E5%A5%B3&pn=50>

<https://tieba.baidu.com/f?kw=%E7%BE%8E%E5%A5%B3&pn=100>

①获取贴吧的URL

<http://tieba.baidu.com/f>? + kw= 美女 + &pn=100

pn = (page-1)\*50

②获取每个帖子的url

//div[@class="col2\_right j\_threadlist\_li\_right "]/div/div/a/@href



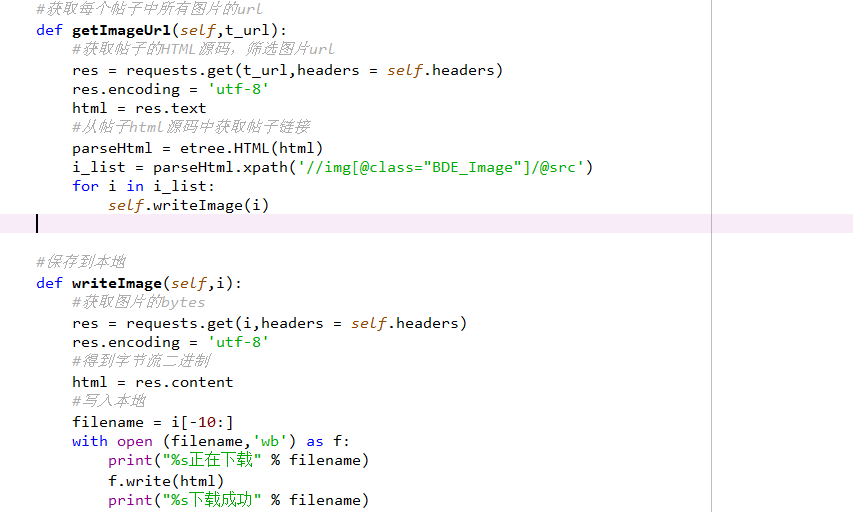
<https://tieba.baidu.com/p/5878541754>

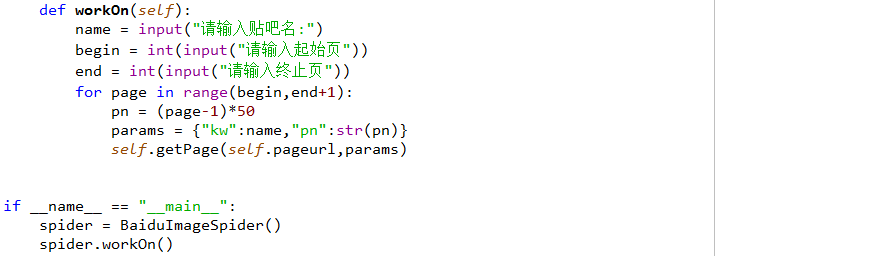
③打开帖子，找到每个图片

//div[@id="post\_content\_121963174249"]/img/@src









5 ProxyBasicAuthHandler私密代理Handler处理器

5.1 密码管理器使用流程

1.创建密码管理器对象

pwd=urllib.request.HTTPPasswordMgrWithDefaultRealm()

2.添加私密代理用户名，密码，IP地址，端口信息

pwd.add\_password(None,”IP:端口”,”用户名”,”密码”)

3.urllib.request.ProxBasicAuthHandler(密码管理器对象)

网络爬虫5

1 糗事百科-Xpath

目标：爬取用户昵称、段子内容，好笑数、评论数

步骤：

1.找URL

<http://www.qiushibaike.com/8hr/page/1/>

2.xpath匹配

基准的xpath每个段子：

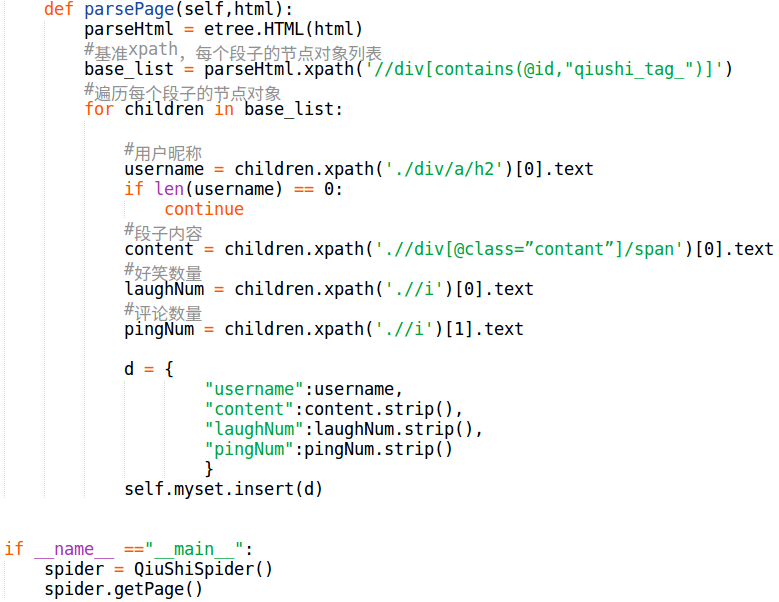
//div[contains(@id,”qiushi\_tag\_”)]

用户昵称：./div/a/h2  
段子内容：.//div[@class=”contant”]/span

好笑数量：

评论数量：





2 动态网站数据抓取 - AJAX

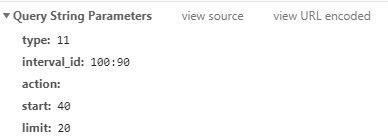
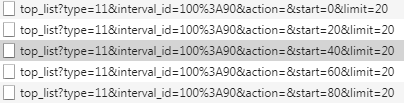
2.1 Ajax动态加载

特点：动态加载（滚动鼠标滑轮时加载）

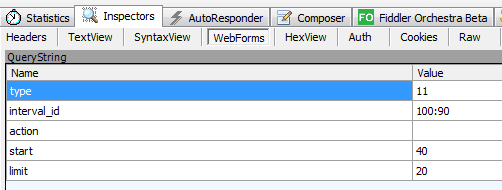
案例：豆瓣电影top100榜单（剧情片）

<https://movie.douban.com/typerank?type_name=%E5%89%A7%E6%83%85&type=11&interval_id=100:90&action>=

谷歌浏览器抓包

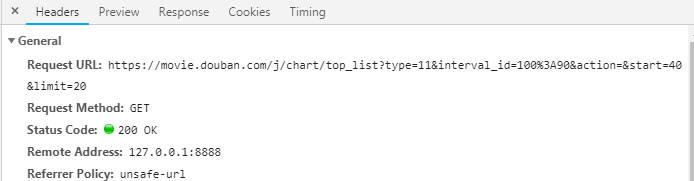


对比Fiddler抓包

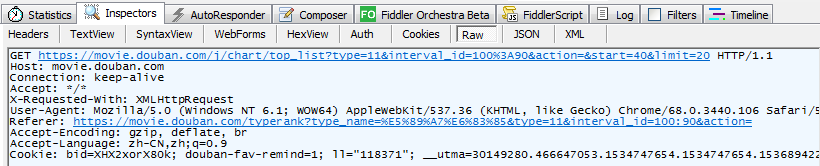


处理为字典

地址要写Request URL里的地址



对比fiddler抓包





3 json模块

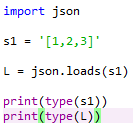
作用：

json格式类型和Python数据类型相互转换

常用方法

json.loads()：json格式-->Python数据类型

|  |  |
| --- | --- |
| json | python |
| 对象 | 字典 |
| 数组 | 列表 |



4 selenium+phantomjs强大的网络爬虫

4.1 selenium

定义：

Web自动化测试工具，应用于web自动化测试

特点：

可运行在浏览器上，根据指令操作浏览器，让浏览器自动加载页面；

只是工具，不支持浏览器功能

安装：

conda install selenium

或pip3 install selenium

验证：

from selenium import webdriver

4.2 phantomjs

定义：

无界面浏览器（无头浏览器）

特点：

1.把网站加载到内存执行页面加载

2.运行高效

安装：

windows

把安装包拷贝到Python安装目录下scripts…

c:\python36\Script

启动phantomjs

在安装目录python36下输入phantomjs

liunx

1.下载phantomjs，放在任意路径

2.在用户主目录下:vi .bashrc

export PHANTOM\_JS=/home/…/phantomjs

export PATH=$PHANTOM\_JS/bin:$PATH

保存

3.source .bashrc

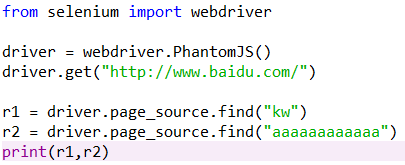
4.终端：phantomjs

4.3 常用方法

1.driver.get(url)

2.driver.page\_source.find(“内容”)：

从html源码中搜索字符串，搜索成功返回非-1，搜索失败返回-1



3.driver.find\_element\_by\_id(“属性值”).text

4.driver.find\_element\_by\_name(“属性值”)

5.driver.find\_element\_by\_class\_name(“属性值”)

6.对象名.send\_keys(“内容”)

7.对象名.click()

8.driver.quit()



4.4 案例：登录豆瓣网站



4.4 案例：斗鱼直播抓取（JS分页加载）

1.抓取目标：主播 观众人数

主播：span-》class="dy-name ellipsis fl"

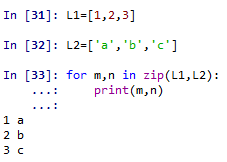
人数：span-》class="dy-num fr"

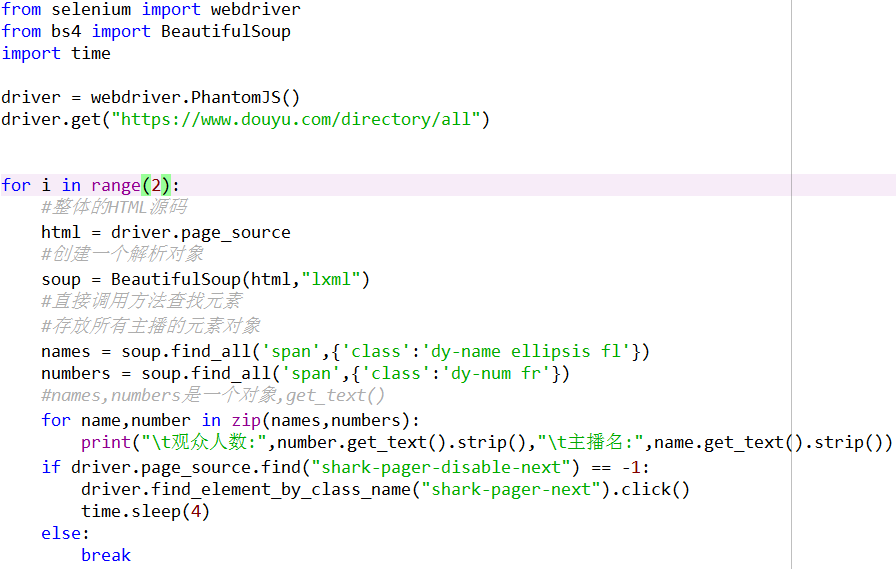
下一页：

能点：a-》class="shark-pager-next"

不能点：a-》

class="shark-pager-nextshark-pager-disableshark-pager-disable-net





5 BeautifulSoup

定义：

HTML或XML的解析器，依赖于lxml库

安装并导入

pip3 install beauitfulsoup4

或conda install beauitfulsoup4

安装模块出错

conda install selenium ---> 报错

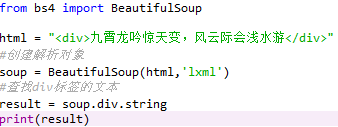
卸载：conda uninstall selenium

安装：cmd 右键->管理员身份打开

python -m pip install selenium

导入模块

from bs4 import BeautifulSoup





5.1 BeaultfulSoup支持解析库

lxml HTML解析器，lxml速度快，文档容错能力强

Python标准库，html.parser速度一般

lxml XML解析器，xml速度